UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 09/921,458 | 08/03/2001 | Jack L. Hong | 4366-40 | 7011 |

48500 7590 12/29/2004

SHERIDAN ROSS P.C.
1560 BROADWAY, SUITE 1200
DENVER, CO 80202

| EXAMINER | |
|---|---|
| CLARK, ISAAC R | |

| ART UNIT | PAPER NUMBER |
|---|---|
| 2154 | |

DATE MAILED: 12/29/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

| Office Action Summary | Application No. | Applicant(s) | |
|---|---|---|---|
| | 09/921,458 | HONG ET AL. | |
| | Examiner | Art Unit | |
| | Isaac R Clark | 2154 | |

*-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

**Period f r Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on *03 August 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) *1-39* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-39* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11/06/2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 04/02/02; 03/04/03; 11/19/04
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

## DETAILED ACTION

1. Claims 1-39 are presented for examination.

### *Priority*

2. The applicant claims priority under 35 USC § 119(e) from Provisional Application No. 60/223087 filed 08/04/2000.

### *Drawings*

3. The Examiner contends that the drawings submitted on 11/06/2001 are acceptable for examination proceedings.

### *Specification*

4. In the section "Cross reference to related applications, the specification makes reference to two copending applications by title (page 1, lines 9-11). The references are incomplete because application numbers are not provided. If a reference has matured into a patent it is additionally required that this fact be incorporated into the present specification in order to reflect accurate information and readily allow identification of related documentation.

### *Double Patenting*

5. A rejection based on double patenting of the "same invention" type finds its support in the language of 35 U.S.C. 101 which states that "whoever invents or discovers any new and useful process ... may obtain a patent therefor ..." (Emphasis added). Thus, the term "same invention," in this context, means an invention drawn to identical subject matter. See *Miller v. Eagle Mfg. Co.*, 151 U.S. 186 (1894); *In re Ockert*, 245 F.2d 467, 114 USPQ 330 (CCPA 1957); and *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970).

A statutory type (35 U.S.C. 101) double patenting rejection can be overcome by canceling or amending the conflicting claims so they are no longer coextensive in scope. The filing of a terminal disclaimer cannot overcome a double patenting rejection based upon 35 U.S.C. 101.

6. Claims 2 and 27 are provisionally rejected under 35 U.S.C. 101 as claiming the same invention as that of claims 2 and 26 of copending Application No. 09/921832. This is a provisional double patenting rejection since the conflicting claims have not in fact been patented.

7. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

8. Claims 9 and 16 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 8 and 17 of copending Application No. 09/921832 in view of Jordan et al. (US Published Application 2002/0026560) hereinafter Jordan.

9. Claims 9 and 16 in the instant application differ from claims 8 and 17 respectively in the copending application in that the claims in the instant application fail to describe selecting between a first server and a cache server based on the stored field information. Jordan in an analogous art teaches maintaining a hit counter and when the hit counter at least one of equals or exceeds a predetermined threshold, transferring content associated with the transaction request from the corresponding server to a

cache server (Fig 3. steps 204 and 205, Paragraph 0030: transferring content to a new cache server).

This is a provisional obviousness-type double patenting rejection.

***Claim Rejections - 35 USC § 102***

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

11. Claims 32, 33, and 35 are rejected under 35 U.S.C. 102(e) as being anticipated by Jordan et al. (US Published Application 2002/0026560) hereinafter Jordan.

12. As per claim 32, Jordan teaches a method for processing selected information in a network packet, comprising: parsing the packet to locate at least one of a universal resource locator and a cookie; determining a hash value from at least a portion of the at least one of a universal resource locator and a cookie (Paragraph 0038: hash performed on URL from the request); and storing information associated with the packet at a location corresponding to the hash value (Paragraph 0038; storing requested object in a cache).

13. As per claim 33, Jordan teaches the method of claim 32, wherein the hash value is based on a hash function (Paragraph 0038).

14. As per claim 35, Jordan teaches the method of claim 32, wherein the information is stored in a cache, wherein the information includes a hit counter related to hotness of content corresponding to the universal resource locator (Fig. 2b; Paragraph 0029; Object Id is based on hash function; Paragraph 0030: hit counter incremented based on a request), and further comprising: determining a server destination for the packet based on the stored information (Paragraph 0038).

15. Claims 36 and 37 are rejected under 35 U.S.C. 102(e) as being anticipated by Apostolopoulos et al., "Design, implementation and performance of a content-based switch", INFOCOM 2000, Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies, pp. 1117-1126, published 26 March 2000 (hereinafter Apostolopoulos).

16. As per claim 36, Apostolopoulos discloses a system, comprising: a communications network (Fig. 3); a plurality of replicated servers connected to the network, all of the replicated servers having a same network address and all of the replicated servers serving the same replicated information (page 122, col. 2, final paragraph), each of the replicated servers being configured to receive a first transaction request associated with an individual transaction and to provide a response to the first transaction request, the response including a tag that corresponds to the transaction (page 1123, col. 1, URL used to map to candidate set of web caches); and a network switch connecting the servers to the network, the network switch being configured to receive all transaction requests addressed to the network address, to select one of the servers to serve the first transaction request (page 1123, col. 1: request directed to least

loaded server), to store the tag that is associated with the transaction in association with a selected server, and to direct to the selected server subsequent received transaction requests including the tag (page 1123, col. 1), wherein the network switch stores the tag at a memory location based on a digest value, the digest value being determined at least in part by at least a portion of the tag (page 1123, col. 2: digest value is hash value of URL).

17. As per claim 37, Apostolopoulos discloses the system of claim 36, wherein the network switch comprises a parser configured to parse the first transaction request and wherein the tag is at least one of a universal resource locator and a cookie (page 1123, col. 1, tag is URL).

***Claim Rejections - 35 USC § 103***

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

19. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to

consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

20. Claims 1-3, 5, 7-9, 14, 16-18, 23, 25-28, and 30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hankinson et al. (US 6,799,202) hereinafter Hankinson in view of Schmeidler et al. (US 6,763,370) hereinafter Schmeidler.

21. As per claim 1, Hankinson teaches a network switch (col. 2, lines 38-40) for switching transaction requests among a plurality of information servers (col. 2, lines 34-35), comprising: a routing component that parses transaction requests to locate selected fields and thereafter forwards at least portions of the parsed transaction requests to respective information servers (col. 2, lines 47-54); a cache that stores a plurality of objects corresponding to transaction requests associated with at least one of the plurality of information servers (col. 15, lines 49-52), the objects comprising field information in at least one of the selected fields located by and received from the routing component.

22. Hankinson does not explicitly teach a digest generator that generates a digest based on the field information in at least one selected field of a corresponding transaction request, the digest corresponding to a location in the cache where at least one object corresponding to the corresponding transaction request is to be stored; and a cache processor that accesses the plurality of objects in response to communications received from the routing component.

23. Schmeidler teaches the network switch further comprising a digest generator that generates a digest based on the field information in at least one selected field of a

corresponding transaction request, the digest corresponding to a location in the cache where at least one object corresponding to the corresponding transaction request is to be stored; and a cache processor that accesses the plurality of objects in response to communications received from the routing component (col. 18, lines 43-51; digest value is a computed hash).

24. It would have been obvious to one of ordinary skill in this art at the time the invention was made to modify Hankinson with the teaching of Schmeidler to generate a hash as a digest to be used as an index because they both deal with client access to data on servers. Furthermore, the teaching of Schmeidler to use a hash as an index would provide efficient access to the cached data structures (Schmeidler, col. 4, lines 35-47).

25. As per claim 2, Hankinson discloses the network switch of claim 1, further comprising a decryption processor that decrypts cipher text transaction requests (col. 8, lines 54-57) and provides plain text transaction requests to the routing component (col. 18, lines 40-48).

26. As per claim 3, Hankinson teaches the network switch of claim 1, further comprising at least one traffic manager located between the network switch and one or more clients (Fig. 10; col. 18, lines 60-67; dispatcher determines which responder to reply to a client).

27. Hankinson does not explicitly teach a digest generated by a hashing function.

28. Schmeidler as applied to claim 2 above teaches that digest is generated by a hashing function. The rationale for combining Hankinson and Schmeidler is as described for claim 2 above.

29. As per claim 5, Hankinson teaches the network switch of claim 1, wherein the routing component includes a current connection table listing active connections between information servers and clients (col. 15, lines 1-6).

30. As per claim 7, Hankinson teaches a method for switching transaction requests among a plurality of information servers (col. 2, lines 38-40), comprising: receiving a transaction request corresponding to an information server and parsing one or more selected fields in the transaction request (col. 2, lines 47-54).

31. Hankinson fails to explicitly teach determining a digest value based on field information in at least one of the selected fields and storing selected information corresponding to the transaction request at an address based on the digest value.

32. Schmeidler teaches a method for switching transaction requests further comprising a digest generator that generates a digest based on the field information in at least one selected field of a corresponding transaction request and storing selected information corresponding to the transaction request at an address based on the digest value. (col. 18, lines 43-51; digest value is a computed hash).

33. It would have been obvious to one of ordinary skill in this art at the time the invention was made to modify Hankinson with the teaching of Schmeidler to generate a hash based on the field information of a request as a digest to be used as an index because they both deal with client access to data on servers. Furthermore, the

teaching of Schmeidler to use a hash as an index would provide efficient access to the cached data structures (Schmeidler, col. 4, lines 35-47).

34. As per claim 8, Hankinson teaches the method of claim 7, wherein the transaction request is in hypertext transfer protocol (col. 11, lines 42-52) and that the field information includes a universal resource locator (col. 11, lines 64-67).

35. Hankinson does not explicitly teach a digest value is generated by a hashing function, and the field information used to determine the digest value is at least one of a universal resource locator and a cookie.

36. Schmeidler teaches a method for switching transaction requests further comprising a digest generator that generates a digest based on the field information in at least one selected field of a corresponding transaction request and storing selected information corresponding to the transaction request at an address based on the digest value. (col. 18, lines 43-51; digest value is a computed hash).

37. It would have been obvious to one of ordinary skill in this art at the time the invention was made to modify Hankinson with the teaching of Schmeidler to generate a hash based on the field information of a request as a digest to be used as an index because they both deal with client access to data on servers. Furthermore, the teaching of Schmeidler to use a hash as an index would provide efficient access to the cached data structures (Schmeidler, col. 4, lines 35-47).

38. As per claim 9, Hankinson teaches the method of claim 7, wherein the transaction request is in cipher text (col. 8, lines 54-57) and further comprising after the

receiving step and before the parsing step decrypting the transaction request (col. 18, lines 40-48; col. 21, line 64 – col. 22, line 4).

39. As per claim 14, Hankinson teaches the method of claim 7, further comprising: determining whether the transaction request can be served by a cache server; and if the transaction request cannot be served by the cache server, forwarding the transaction request to the corresponding server (col. 15, lines 49-51: server stores data in cache; col. 17, lines 40-55: determining which server to handle the transaction the transaction request).

40. As per claim 16, claim 16 describes an apparatus for carrying out the method described in claim 7. Claim 7 is rejected for the same reasons as claim 7 above.

41. As per claim 17, claim 17 describes an apparatus for carrying out the method described in claim 8. Claim 8 is rejected for the same reasons as claim 7 above.

42. As per claim 18, Hankinson teaches the system of claim 16, wherein the transaction request is in cipher text and further comprising between the input port and the parsing means (col. 8, lines 54-57) decrypting means for decrypting the transaction request (col. 18, lines 40-48; col. 21, line 64 – col. 22, line 4).

43. As per claim 23, Hankinson teaches the system of claim 16, further comprising: second determining means for determining whether the transaction request can be served by a cache server; and if the transaction request cannot be served by the cache server, forwarding means for forwarding the transaction request to the origin server (col. 15, lines 49-51: server stores data in cache; col. 17, lines 40-55: determining which server to handle the transaction the transaction request).

44. As per claim 25, Hankinson teaches a network switch (col. 2, lines 38-40) for switching transaction requests among a plurality of servers (col. 2, lines 34-35), comprising: a cache that stores a plurality of objects corresponding to transaction requests associated with at least one of a plurality of servers, the objects comprising field information in selected fields in the transaction requests (col. 15, lines 49-52); and a cache processor that accesses the plurality of objects (col. 15, lines 49-52).

45. Hankinson does not explicitly teach a digest generator that generates a digest value based on the field information in at least one selected field of a corresponding transaction request, the digest value corresponding to a location in the cache where at least one object corresponding to the transaction request is to be stored

46. Schmeidler teaches a network switch including a digest generator that generates a digest value based on the field information in at least one selected field of a corresponding transaction request, the digest value corresponding to a location in the cache where at least one object corresponding to the transaction request is to be stored (col. 18, lines 43-51; digest value is a computed hash).

47. It would have been obvious to one of ordinary skill in this art at the time the invention was made to modify Hankinson with the teaching of Schmeidler to generate a hash as a digest to be used as an index because they both deal with client access to data on servers. Furthermore, the teaching of Schmeidler to use a hash as an index would provide efficient access to the cached data structures (Schmeidler, col. 4, lines 35-47).

48. As per claim 26, Hankinson teaches the network switch of claim 25, further comprising: a routing component that parses the transaction requests to locate the selected fields and thereafter forwards at least portions of the parsed transaction requests to respective servers (col. 2, lines 47-54) and wherein the cache processor accesses the plurality of objects in response to communications received from the routing component (col. 15, lines 49-54).

49. As per claim 27, Hankinson teaches the network switch of claim 26, further comprising a security processor that decrypts cipher text transaction requests (col. 8, lines 54-57) and provides plain text transaction requests to the routing component (col. 18, lines 40-48).

50. As per claim 28, Hankinson teaches the network switch of claim 26, further comprising at least one traffic manager located between the network switch and one or more clients (Fig. 10; col. 18, lines 60-67; dispatcher determines which responder to reply to a client). Hankinson does not explicitly teach that the digest value is generated using a hashing function.

51. Schmeidler teaches that the digest value is generated by using a hashing function (col. 18, lines 43-51; digest value is a computed hash). The rationale for combining Hankinson and Schmeidler is the same as for claim 25 above.

52. As per claim 30, Hankinson teaches the network switch of claim 26, wherein the routing component includes a current connection table listing active connections between servers and clients (col. 15, lines 1-6).

53. Claims 4 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hankinson and Schmeidler as applied respectively to claims 1 and 26 above, and further in view of Masters (US Published Application 2002/0040400).

54. As per claim 4, Hankinson teaches the network switch of claim 1 where the selected fields include a resource locator (col. 11, lines 42-52 and 64-67). Hankinson teaches that additional that additional data in the request may be selected, but does not explicitly teach that the data includes a cookie.

55. Masters teaches parsing information from a cookie in a HTTP request as part of the index into a table of destination servers (Paragraph 0017)

56. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Masters to include a cookie in the selected field information because they both deal with balancing loads among a plurality of servers. Furthermore, the teaching of Masters to include cookie information in the fields used to select the server would allow the load balancing system to support persistent connections in which the client must be returned to the same server upon repeated requests (See Masters Paragraph 0007).

57. As per claim 29, Hankinson teaches a network switch of claim 26 wherein the selected fields include a resource locator (col. 11, lines 42-52 and 64-67). Hankinson teaches that additional that additional data in the request may be selected, but does not explicitly teach that the data includes a cookie.

58. Masters teaches parsing information from a cookie in a HTTP request as part of the index into a table of destination servers (Paragraph 0017)

59. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Masters to include a cookie in the selected field information because they both deal with balancing loads among a plurality of servers. Furthermore, the teaching of Masters to include cookie information in the fields used to select the server would allow the load balancing system to support persistent connections in which the client must be returned to the same server upon repeated requests (See Masters Paragraph 0007).

60. Claims 6 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hankinson and Schmeidler as applied respectively to claim 1 and 26 above, and further in view of Logue et al. (US 6,330,606) hereinafter Logue.

61. As per claim 6, Hankinson discloses the network switch of claim 1 including storing information associated with a packet in a cache (col. 15, lines 49-52).

62. Hankinson does not explicitly teach that the plurality of objects in the cache include a plurality of content addresses for specific content and a corresponding hit counter showing a number of instances in a predetermined period of time in which specific content is requested by transaction requests.

63. Logue teaches storing a plurality of content addresses for specific content and a corresponding hit counter showing a number of instances in a predetermined period of time in which specific content is requested by transaction requests (Fig. 5A; col. 6, lines 35-55; col. 7 lines 15-22; timestamp to designate period covered; col. 9, lines 35-41 periodic resetting of hit counter).

64. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Logue to modify the cached information to include a hit counter for a plurality of addresses requested over a predetermined time interval because they both deal with selecting servers to deliver requested content. Furthermore, the teaching of Logue to provide a hit counter for requested content over a predetermined time interval would allow providing a detailed report of content accessed on a server allowing site owners to judge the popularity and effectiveness of their content (See Logue col. 7, lines 48-59).

65. As per claim 31, Hankinson discloses the network switch of claim 26 including storing information associated with a packet in a cache (col. 15, lines 49-52).

66. Hankinson does not explicitly teach that the plurality of objects in the cache include a plurality of content addresses for specific content and a corresponding hit counter showing a number of instances in a predetermined period of time in which specific content is requested by transaction requests.

67. Logue teaches storing a plurality of content addresses for specific content and a corresponding hit counter showing a number of instances in a predetermined period of time in which specific content is requested by transaction requests (Fig. 5A; col. 6, lines 35-55; col. 7 lines 15-22; timestamp to designate period covered; col. 9, lines 35-41 periodic resetting of hit counter).

68. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Logue to modify the cached information to include a hit counter for a plurality of addresses requested over a

predetermined time interval because they both deal with selecting servers to deliver requested content. Furthermore, the teaching of Logue to provide a hit counter for requested content over a predetermined time interval would allow providing a detailed report of content accessed on a server allowing site owners to judge the popularity and effectiveness of their content (See Logue col. 7, lines 48-59).

69. Claims 10-12, 15, 19-21, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hankinson and Schmeidler as applied to claim 7 above, and further in view of Jordan et al. (US Published Application 2002/0026560) hereinafter Jordan.

70. As per claim 10, Hankinson fails to explicitly teach the method of claim 7, wherein storing step comprises: at least one of incrementing and decrementing a hit counter; determining if the hit counter at least one of equals or exceeds a predetermined threshold when the hit counter is incremented or at least one of equals or is less than the predetermined threshold when the hit counter is decremented; and updating a timestamp associated with the stored information.

71. Jordan teaches incrementing a hit counter, determining if the hit counter exceeds a predetermined threshold when the hit counter is incremented and updating a timestamp associated with the stored information (Fig. 3; Paragraph 0029: forwarding is request counter; updating time stamp; Paragraph 0030: determining that hit counter has exceeded threshold).

72. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Jordan to include as part of storing information concerning a cache request, incrementing a counter and

determining when the counter exceeds a predetermined threshold because they both with selecting a server to handle a transaction request. Furthermore, the teaching of Jordan to maintain a hit counter and a time stamp associated with an information request would provide a technique for improving load balancing by identifying popular requests for information.

73. As per claims 11 and 12, Hankinson fails to explicitly teach the method of claim 10, wherein when the hit counter at least one of equals or exceeds the predetermined threshold, determining a plurality of network addresses associated with content referenced in the transaction request and directing the transaction request to a cache server that is different from an origin server corresponding to the transaction request.

74. Jordan teaches maintaining a hit counter and when the hit counter equals or exceeds the predetermined threshold, determining a plurality of network addresses associated with content referenced in the transaction request and directing the transaction request to a cache server that is different from an origin server corresponding to the transaction request (Fig. 3: steps 204 and 205, Paragraph 0030; assigns new owner based on overloaded condition).

75. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Jordan to include as part of storing information concerning a cache request, incrementing a counter and determining directing the transaction request to a cache server when the counter equals or exceeds a predetermined value. Furthermore, the teaching of Jordan to maintain a hit counter and provide a cache server when the counter exceeded a predetermined

value would provide improved load balancing by identifying popular requests for information and determining cache content based on the identified requests (See Jordan Paragraph 0011).

76. As per claim 15, Hankinson fails to explicitly teach the method of claim 7, further comprising when a hit counter at least one of equals or exceeds a predetermined threshold, transferring content associated with the transaction request from the corresponding server to a cache server.

77. Jordan teaches maintaining a hit counter and when the hit counter at least one of equals or exceeds a predetermined threshold, transferring content associated with the transaction request from the corresponding server to a cache server (Fig 3. steps 204 and 205, Paragraph 0030: transferring content to a new cache server).

78. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Jordan to include as part of storing information concerning a cache request, incrementing a counter and transferring content associated with the request to a cache server when the counter equals or exceeds a predetermined value. Furthermore, the teaching of Jordan to maintain a hit counter and provide a cache server when the counter exceeded a predetermined value would provide improved load balancing by identifying popular requests for information and determining cache content based on the identified requests (See Jordan Paragraph 0011).

79. As per claims 19, 20, 21, and 24, claims 19, 20, 21, and 24 describe an apparatus for carrying out the method steps described in claims 10, 11, 12, and 15 respectively, and are rejected for the same reasons as claims 10, 11, 12, and 15.

80. Claims 13 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hankinson and Schmeidler as applied respectively to claim 7 and 16 above further in view of Masters (US Published Application 2002/0040400) and Jordan et al. (US Published Application 2002/0026560) hereinafter Jordan.

81. As per claim 13, Hankinson fails to teach the method of claim 7, further comprising: determining whether the transaction request is a part of an existing connection between the origin server and a client; when the transaction request is part of an existing connection, forwarding the transaction request to the corresponding server; and when the transaction request is not part of an existing connection and a hit counter at least one of equals or exceeds a predetermined value, forwarding the transaction to a cache server different from the corresponding server.

82. Masters teaches determining whether the transaction request is a part of an existing connection between the origin server and a client and when the transaction request is part of an existing connection, forwarding the transaction request to the corresponding server and when the transaction request is not part of an existing connection using a load balancing scheme to direct the transaction request (Paragraph 0050 and 0051: use of cookie to identify that request is part of an existing connection).

83. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Masters to provide a

network switch where transaction requests that are part of the same connection are directed to the same server. Furthermore, the teaching of Masters to include cookie information in the fields used to select the server would support efficient persistent connections by avoiding the need to transfer state relationship information between servers (See Masters Paragraph 0050).

84. Jordon teaches a load balancing scheme including a hit counter including when the hit counter at least one of equals or exceeds a predetermined value forwarding the transaction to a cache server different from the corresponding server (Fig. 3, steps 203-206; Paragraph 0030).

85. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Johnson to provide a network switch where when transaction requests are not part of the same transaction, and when a hit counter exceeds a predetermined value the transaction is forwarded to a cache server different from the corresponding server. Furthermore, the teaching of Jordon to maintain a hit counter and provide a cache server when the counter exceeded a predetermined value would provide improved load balancing by identifying popular requests for information and determining cache content based on the identified requests (See Jordan Paragraph 0011).

86. As per claim 22, Hankinson fails to teach the system of claim 16, further comprising: second determining means for determining whether the transaction request is a part of an existing connection between the origin server and a client; when the transaction request is part of an existing connection, forwarding means for forwarding

the transaction request to the origin server; and when the transaction request is not part of an existing connection and a hit counter at least one of equals or exceeds a predetermined value, the forwarding means forwards the transaction to a cache server different from the origin server.

87. Masters teaches determining whether the transaction request is a part of an existing connection between the origin server and a client and when the transaction request is part of an existing connection, forwarding the transaction request to the corresponding server and when the transaction request is not part of an existing connection using a load balancing scheme to direct the transaction request (Paragraph 0050 and 0051: use of cookie to identify that request is part of an existing connection).

88. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Masters to provide a network switch where transaction requests that are part of the same connection are directed to the same server. Furthermore, the teaching of Masters to include cookie information in the fields used to select the server would support efficient persistent connections by avoiding the need to transfer state relationship information between servers (See Masters Paragraph 0050).

89. Jordon teaches a load balancing scheme including a hit counter including when the hit counter at least one of equals or exceeds a predetermined value forwarding the transaction to a cache server different from the corresponding server (Fig. 3, steps 203-206; Paragraph 0030).

90. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Johnson to provide a network switch where when transaction requests are not part of the same transaction, and when a hit counter exceeds a predetermined value the transaction is forwarded to a cache server different from the corresponding server. Furthermore, the teaching of Jordon to maintain a hit counter and provide a cache server when the counter exceeded a predetermined value would provide improved load balancing by identifying popular requests for information and determining cache content based on the identified requests (See Jordan Paragraph 0011).

91. Claim 34 is rejected under 35 U.S.C. 103(a) as being unpatentable over Jordan et al. (US Published Application 2002/0026560) hereinafter Jordan in view of Ross, K., "Hash-Routing for Collections of Shared Web Caches," November/December 1997 IEEE Network Magazine, pp. 37-44 (hereinafter Ross).

92. As per claim 34, Jordan teaches the method of claim 31, wherein a universal resource locator is used in the hash function to determine the hash value (Paragraph 0038).

93. Jordan fails to explicitly teach using only a portion of the universal resource locator is used in the hash function to determine the hash value.

94. Ross teaches using a portion of the universal resource locator is in the hash function to determine the hash value (See page 40, col. 1, final paragraph).

95. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Jordan and Ross to store information

associated with a packet using only a portion of the universal resource locator because they both selecting a cache to service a request. Furthermore, the teaching of Ross would result in caching an entire web site at single server resulting in a reduced number of connections and supporting persistent connections in which subsequent requests must go to the same server (See Ross page 40, col.1 final paragraph).

96. Claim 38 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos as applied to claim 37 above, and further in view of Hankinson et al. (US 6,799,202) hereinafter Hankinson.

97. As per claim 38, Apostolopoulos fails to explicitly teach the system of claim 37, wherein the network switch comprises a decryption processor configured to decrypt the first transaction request before the parser parses the first transaction request.

98. Hankinson teaches the system wherein the network switch comprises a decryption processor (Fig. 7, item 735) configured to decrypt a transaction request before the parser (Fig. 7, item 725) parses the transaction request (col. 22, lines 1-9).

99. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Hankinson and Apostolopoulos to modify the network switch system of Apostolopoulos to decrypt an encrypted transaction request prior to making parsing the request because they both selecting a server to process a client transaction request. Furthermore, the teaching of Hankinson to decrypt the transaction request reduces the traffic required by allowing switching based on the content of an encrypted packet by allowing related SSL sessions to be sent to the same server (See Apostolopoulos, page 1117, col. 2, first full paragraph).

100. Claim 39 is rejected under 35 U.S.C. 103(a) as being unpatentable over Apostolopoulos as applied to claim 37 above, and further in view of Logue et al. (US 6,330,606) hereinafter Logue.

101. As per claim 39, Apostolopoulos fails to explicitly teach the system of claim 37 wherein the tag is part of a plurality of stored objects and the plurality of stored objects correspond to the first transaction request and wherein the plurality of stored objects include a hit counter indicating a frequency of transaction requests for content associated with the first transaction request.

102. Logue teaches storing a plurality of content addresses for specific content and a corresponding hit counter showing a number of instances in a predetermined period of time in which specific content is requested by transaction requests (Fig. 5A; col. 6, lines 35-55; col. 7 lines 15-22; timestamp to designate period covered; col. 9, lines 35-41 periodic resetting of hit counter).

103. It would have been obvious to one of ordinary skill in this art at the time the invention was made to combine the teaching of Apostolopoulos and Logue to modify the cached information to include a hit counter for a plurality of addresses requested over a predetermined time interval because they both deal with selecting servers to deliver requested content. Furthermore, the teaching of Logue to provide a hit counter for requested content over a predetermined time interval would allow providing a detailed report of content accessed on a server allowing site owners to judge the popularity and effectiveness of their content (See Logue col. 7, lines 48-59).

***Conclusion***

104. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following patents and publications are cited to further show the state of the art with respect to "Intelligent demand driven recognition of URL objects in connection oriented transactions".

i. US 6,134,583 Herriot Cache table with URL tag and pointer to server

ii. Alteon Web Systems, "Virtual Matrix Architecture, Scaling Web Services for Performance and Capacity" April 2000. (Provided on Applicant IDS).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Isaac R Clark whose telephone number is (571)272-3961. The examiner can normally be reached on Monday-Friday 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John A Follansbee can be reached on (571)272-3964. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

IRC

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100